Generic and Fully Automatic Content-Based Image Retrieval Using Color

Suresh K. Choubey* Vijay V. Raghavan†

Abstract

In this paper, we propose a generic and efficient content-based image retrieval architecture. We compute "real" interimage distances for an initial subset of the images that are to be stored into an image database. For computing real interimage distances we use image content based on a low level feature. High-level image feature vectors are computed from the real interimage distances in such a way that the interimage distances are preserved in the feature space defined by the high-level features. These feature vectors are used to represent the images in the initial subset as well as to generate a training set. This training set is used to compute the feature vector of a query image during image retrieval and for deriving the feature vectors for images not in the initial subset. On-line retrieval is performed using the distances of the feature vector of the query image to feature vectors of the database images as estimates of the corresponding real distances. We have conducted experiments using color as the low level feature. Our results show a substantial reduction in the size of the feature space, which leads to highly efficient on-line retrieval. We also demonstrate that a high retrieval accuracy is achieved.

Keywords: Image database, image retrieval, color, efficient, generic.

1 Introduction

A *content-based image retrieval* (CBIR) system uses information from the content of images for retrieval and helps the user retrieve images relevant to the contents of a query image. For a given query image, its content is directly compared with that of the images in the image database and a desired number of images close to the query image are retrieved. By content, we mean computable and low level features, such as color, shape, texture, object centroids and boundaries. It is very difficult to extract semantics associated with a given image [1].

Approaches to CBIR can be classified into two broad classes of *attribute-based* and *feature-based*. In the attribute-based approach, the image contents are modeled as a set of attributes extracted manually or semi-automatically and managed within the framework of conventional DBMS [1]. Images are represented at a very high-level of abstraction. Queries are also specified using these attributes. Due to the high level of abstraction in image representation, it is difficult to confront the image database with ad hoc queries such as "Give me all images with red color" or "Give me all images of airplane-like shape".

In a feature-based CBIR, images are represented by their contents and the comparison is made between the contents of the query and the images in the image database. This approach is very computation intensive and may not be really effective with respect to response time. The *feature-based* approach can be further classified based on the dependence of the retrieval process on the particular low level feature selected and the degree of automation. In our context, a generic system is defined as one, where the processing steps remain largely the same for different choices of image properties. Approaches to CBIR can be either *semi-automatic non-generic*, *semi-automatic generic*, *automatic non-generic*, or *automatic generic*. The ultimate goal of the *feature-based* approach is a *generic and fully automatic* approach. In such an approach, feature extraction and indexing of images based on features are done with almost no human involvement to get an acceptable response time and the use of various low level image properties does not require any change in the retrieval process.

We have reported image retrieval specific to texture only in [2]. We have also described a software architecture, in [3], for a generic and efficient feature-based CBIR using color, shape, and texture properties. In this paper, we experimentally evaluate our approach using color as the low level image property.

This paper is organized into six sections. Section 2 discusses related work and our motivation. In the next section, we define the problem formally. Sections 4 and 5 cover our approach and experimental work. The results are summarized in section 6.

*choubey@mr.med.ge.com, Magnetic Resonance Imaging, GE Medical Systems, Waukesha, WI 53188, Phone: 414-521-6998 FAX: 414-521-6439

†raghavan@cacs.usl.edu, The Center for Advanced Computer Studies, University of SW Louisiana,Lafayette, LA 70504, USA, Phone: 318-482-6603 Fax: 318-482-5791

2 Related Work and Motivation

A review of the literature [4, 5] shows that color has been used extensively as a discriminating feature in IR problems. However, the use of "real" distances using color in IR problems is computationally very expensive, since it has to account for the effect of color correlations.

Chang et al. [4] used color as a discriminating feature for image retrieval. For a given query, the images in the image database are ranked on the basis of real distance between the query image and the images in the image database. Hence the on-line retrieval is very computationally intensive. Faloutsos et al. [5], while using low level features such as color, have tried to make it more efficient by using estimated distance instead of real distance. They reduce the search space by eliminating many images from consideration by using an estimated distance in such a way that the lower bound lemma is satisfied. This lemma does not allow the elimination of relevant images, but may retain many non-relevant images. Then they perform the search on the basis of real distance in the search space with fewer images. Also, the use of estimated distance does not usually preserve the order induced by the real distance among the images. Hence, they cannot perform nearest neighbor type searches. Some other researchers have tried to make the retrieval efficient at the cost of accuracy by reducing the image resolution [6]. However, when images are quantized to a lower resolution, there is a very strong possibility that some of the important color information relevant for differentiation is lost.

Our proposed method differs from earlier work with respect to the following: We identify a subset of all images, called the initial image set, and use real distances among images in this set to compute high-level image features and to generate a training set. We use the training set to compute the feature vector corresponding to a query image. This requires on-line computation of real distances from the query image to only the images in the training set. In contrast, Faloutsos et al. [5] compute real distances of a query image to all images that are not filtered out by means of their estimated distances. Our estimated distance is more accurate compared to the estimated distance used by Faloutsos et al. [5] in the sense that the ordering of images corresponding to real distances is better preserved by the proposed method for estimating distances.

3 Problem Definition

Let us assume that an image database is populated with a set (or a collection) of images $\{O_0, O_1, \ldots, O_{n-1}\}$. Let this collection be denoted by U. Let Q be a query image.

Let the real interimage distance Π, between any two images O_i, and O_j, one of which may be Q, be denoted by $\Pi(O_i, O_j)$. The user can specify a query to retrieve a number of relevant images. Let q be the number of images, closest to the query image Q, that the user wants to retrieve, assuming that $q < n$.

This image retrieval problem can be defined as the *efficient* retrieval of the best q images according to Π from a database of n images.

4 Proposed Method

This is based on the work of Goldfarb [7] to bridge the gap between syntactic and statistical pattern recognition for classification problems. The whole image database is divided into an *initial* and an *incremental* image set. The complete image storage and retrieval involves two phases: database population and image retrieval. The database population phase uses low level image properties such as color to compute the real interimage distances for the images of initial image set. The size of the initial image set is expected to be much smaller than that of the image database itself. Starting from the interimage distances the high-level image feature vectors for these images are computed and at the same time the training set is generated (see section 3.2). These feature vectors have considerably fewer features compared to an intermediate representation of images based on low level image properties and the extent to which they preserve the real interimage distances among the images, can be controlled. The training set has even fewer images compared to the initial image set. Images from the incremental image set are added to the database using their real distances to only the images in this training set. In the image retrieval phase, we obtain the low level representation of a query image, compute its interimage distance from the images in the training set, compute the query feature vector, and perform retrieval according to the Euclidean distances between this query feature vector and the feature vectors of the database images.
[Figure 1 should be inserted here]

Our architecture (Figure 1) is unique with respect to the generation of the training set during the database population stage. All the possible low level image processing work of feature extraction is performed *a priori* at data population time. Only the query processing is done on-line when the query image becomes available. This leads to an effective, efficient, and flexible image retrieval system.

4.1 Theoretical Background

Let P be the subset of images that has been sampled from the set U to be representative of all possible image classes. This set of images is referred to as the initial set. Let $|P| = k, where \;\; k \leq n$. The value of k should be as small as possible, in order to achieve efficient training.

According to Goldfarb [7], the dissimilarity between two objects can be defined as follows: Let a pseudometric space be a pair (P, Π), where P is a set of images and Π is a non negative real-valued mapping:

$$\Pi : P \times P \rightarrow R^{+} \qquad (1)$$

satisfying the following two conditions:
(a) $\forall O_1 \in P, \forall O_2 \in P \quad \Pi(O_1, O_2) = \Pi(O_2, O_1)$
(b) $\forall O \in P \quad \Pi(O, O) = 0$

The mapping Π is called a pseudometric (or interimage distance) function. This interimage distance will be used from this point on and is the only information about images that will be required in the rest of the image retrieval process. If the number of images in the initial set is k ($i.e., |P| = k$), then the interdistance matrix D is defined as

$$D = (\Pi_{ij})_{0 \le i,j \le k-1}, \qquad (2)$$

Π_{ij} in the above equation is equal to $\Pi(\beta(O_i), \beta(O_j))$. Here β, a vector representation of low level image features, is defined as

$$\beta : P \rightarrow \varphi, \qquad (3)$$

where φ is the space of possible representations of low level image features. For a given $O \in P, \beta\ (O)$ is called the intermediate representation of O.

According to Goldfarb [7], the following definitions lead to a theorem proposed by him that lays the condition for preserving the interimage distance of an interdistance matrix into the derived feature space.

Definition 1: A pair of non-negative numbers (p, q) will be called the *vector signature* of a finite pseudometric space (P, Π), if there exists an isometric embedding

$$\alpha : (P, \Pi) \rightarrow R^{(p,q)} \qquad (4)$$

where $\Pi(\beta(O_1), \beta(O_2)) = \| \alpha(O_1) - \alpha(O_2) \|_2$ $\forall O_1, O_2 \in P$, such that for any other similar isometric embedding of (P, Π) into $R^{(n_1,n_2)}$, we have $n_1 \ge p, n_2 \ge q$. The α is called a *vector representation* of (P, Π).

In other words, (p, q) is the vector signature of a finite pseudometric space (P, Π), if $R^{(p,q)}$ is a minimal pseudo-Euclidean vector space, within which (P, Π) can be isometrically represented. This idea is illustrated in [3]. Isometric embedding ensures that there exists a distance preserving mapping [7]:

Definition 2: Let (P, Π) be the pseudometric space defined earlier and let V be a vector space over R of dimension $k - 1$, and let $\{a_i\}_{1 \le i \le k-1}$ be a basis of such a vector space. A quadratic form on this vector space is given by:

$$\Psi(x) = \sum_{i,j=1}^{k-1} \frac{1}{2}(\Pi_{0i}^2 + \Pi_{0j}^2 - \Pi_{ij}^2)x^i x^j, \qquad (5)$$

for $x = (x^1, \ldots, x^{k-1})$. Here x^i and x^j are the coordinates of x with respect to the standard basis $\{a_i\}$.

Theorem 1: A finite pseudometric space (P, Π) has the vector signature (p, q), iff the quadratic form given in definition 2 has the signature (p, q).

Determining whether a quadratic form has a desired signature requires the use of results on symmetric bilinear forms on (P, Π) and their connection to quadratic forms. The proof of Theorem 1 is given in [8]. A corollary of the theorm 1 is: A finite pseudometric space (P, Π) can be isometrically represented in the Euclidean $m(m = p + q)-$dimensional space iff the quadratic form given in Definition 2 is positive and of rank less than m.

4.2 Database Population Phase

Given the interimage distance matrix, ComputeFeatureVector procedure (Figure 2) is used to compute the high-level image feature vectors for images in the initial set. Let these feature vectors be denoted by $F_0, F_1, \ldots, F_{k-2}, F_{k-1}$. This procedure also generates a training set by calling GenerateTrainingSet() (Figure 3). The training set is used in the computation of feature vectors for a query image and for images in the incremental image set.

4.3 Estimation of Distances and Image Retrieval

Given the query image Q, one can readily determine the orthogonal projection of Q onto the vector representation space $R^{(p,q)}$. ComputeQueryVector() (Figure 4) computes the feature vector of the query image. Feature vectors of the images in the incremental image set, which are also computed using the same algorithm, are denoted by $F_k, F_{k+1}, \ldots, F_{n-1}$.

procedure ComputeFeatureVector (distance matrix D, threshold value tr)

1. From $D = (\Pi_{ij})_{1 \le i,j \le k-1}$, compute $d = \frac{1}{k^2}\sum_{i,j=1}^{k-1} \Pi_{ij}^2$
2. Compute symmetric matrix B in the quadratic form as $b_{ij} = \frac{1}{2}\left[\frac{1}{k}(\sum_{i=1}^{k-1}\Pi_{ij}^2 + \sum_{j=1}^{k-1}\Pi_{ij}^2) - \Pi_{ij}^2 - d\right]$
3. Compute the eigenvalues (characteristic values) of distance matrix B (*QR-Algorithm*).
4. Determine p and q such that p is the # of eigenvalues $> tr$ and q is the # of eigenvalues $< -tr$
5. Let $c_1, c_2, \ldots\ldots, c_p, c_{p+1}, \ldots\ldots\ldots, c_{p+q}, 0, 0, 0, \ldots\ldots, 0$ be the characteristic values of matrix B, such that $c_1 \ge c_2 \ge \ldots \ge c_p > 0$ and $c_{p+1} \le c_{p+2} \le .. \le c_{p+q} < 0$. They form the diagonal matrix C of size $k - 1 \times k - 1$.
6. Determine the corresponding orthogonal characteristic vectors (eigenvectors) of matrix B. They are $e_1, e_2, \ldots\ldots, e_k$. Then the matrix E is formed such that e_i are the columns of matrix E. Can keep only $m = (p+q)$ columns.
7. Compute the matrix M such that $M = E \times C$. The matrix M contains the components of the vector, denoted

as F_i, corresponding to O_i in the pseudo Euclidean vector space $R^{(p,q)}$ under the constructed vector representation $\alpha : (P, \Pi) \rightarrow R^{(p,q)}$

8. Use first m elements from i^{th} column of matrix M to represent image O_i in the pseudo Euclidean vector space $R^{(p,q)}$.
9. call GenerateTrainingSet()

endprocedure

Algorithm to Compute Feature Vectors

procedure GenerateTrainingSet ()

1. Compute the Gram matrix G from F_i, $1 \leq i \leq m$ by using expression $G = A_0^T A_0$.
2. Compute G^{-1} from Gram matrix G.
3. Compute the training set T by multiplying the basis set A_0 by the inverse of Gram matrix. That is, $T = A_0 G^{-1}$.
4. Return T.

endprocedure

Algorithm to Generate Training Set

Since the matrix T can easily be computed during the database population phase, the only on-line computations are those of $b_j, 1 \leq j \leq m$. This is perfectly feasible since one has control over the dimension m of the representation space. Once the query feature vector F_q is computed, the estimated distances are given by the Euclidean distance between the query feature vector and the image vectors.

Procedure ComputeQueryVector(Query Image , Image O, Matrix T)

1. $b_j = \frac{1}{2}\left[\Pi^2(\beta(O), \beta(O_0)) + \Pi^2(\beta(O_j), \beta(O_0))\right]$ $-\frac{1}{2}\left[\Pi^2(\beta(O), \beta(O_j))\right]$ such that $\beta : P \rightarrow \varphi$, where φ is the space of possible representations of low level image features.
2. Determine the orthogonal projection w of O onto the vector representation space $R^{(p,q)}$.
   such that $w = T \cdot b$, and where $b = (b_1, ..., b_m)$
3. return vector w as the feature vector of the new image.

endprocedure

Algorithm to Compute Query Feature Vector

It may be noted that the estimated interimage distance is computed from the high-level feature vectors of images and not the image representations. Also, the computation is very inexpensive, as the feature vectors have a much smaller number of elements. The estimated interimage distance *does* preserve the real interimage distance between two images. It can be used instead of the real interimage distance for efficient image retrieval.

5 Experiments and Results

Two sets of experiments (EXPT I and EXPT II) were conducted using color images from the Department of Water Resources (DWR) repository, maintained at University of California, Berkeley. The purpose of the first set of experiments was to measure the retrieval efficiency and retrieval accuracy of the image database system. The second set of experiments was carried out to study the effect of the threshold value that we use to determine the number of high-level image features to be retained (i.e. the dimensionality of the derived feature space).

We used the RGB color space and a uniform quantization method to quantize the images. Most of the display devices use RGB display and uniform quantization is a simple method and a good choice in the absence of information regarding the color distribution of the images. Uniform quantization can be used in RGB color space very efficiently. Some of the current image retrieval by color use single color, color pairs, and color histograms [4]. Histograms have been widely used in content-based image retrieval using color properties [4, 5, 6] and is considered to be very effective, since it provides robustness with respect to scaling, orientation, perspective, and occlusion.

We used an interimage distance function for retrieval by image color content, also used by Faloutsos et al., Chang el al., and other researchers [4, 5]. The *real interimage distance* function $\Pi(X, Y)$ between two images X and Y, can be computed by using equation 6. The equation 6 can also be written as equation 7.

$$\Pi(X, Y) = (X - Y)^t A (X - Y) \qquad (6)$$

$$\Pi(X, Y) = \sum_{i,j} (x_i - y_i) a_{ij} (x_j - y_j), \qquad (7)$$

where the elements a_{ij} represent the cross correlation between color i and color j, $for \quad 0 \leq i, j \leq L - 1$, and $X = \beta(O_X), Y = \beta(O_Y)$. This compensation takes into account the fact that colors are usually not orthogonal. The commonly used value of L is 64 or 256. This distance function gives the *real* interimage distance between two images O_X and O_Y [4].

In EXPT I and II, sets of 14 and 32 images were used. The images were numbered from 0 through 13 in EXPT I and 0 through 31 in EXPT II. The original images were in Jpeg format. After decompression, these images were quantized to 64 colors and histogram representations for each of the images were derived. From the histograms, the color correlation matrix and the real interimage distances for all pairs of images were computed using equation 7. From these interimage distances (interimage matrix), the high-level image feature vectors were computed. Then ultimately, the estimated interimage distance were derived for each image pair. In this experimental study, all images were used in the initial image set during the database population stage. That is, the incremental image set is empty.

We used the R_{norm} performance measure [9]. This measure is denoted as $R_{norm}(\delta^{estreal})$ in our context and is a way of comparing an ordering of images by estimated distances relative to an ordering of images by real distances. It reaches the maximum value of 1 if, for all $O_i, O_j \in P$, O_i being at a higher rank than O_j by real distances implies that O_i is placed at a higher rank than O_j according to estimated distances. It should be noted that the implication is not both ways. A more detailed discussion of R_{norm} can be found in [9].

5.1 EXPT I Results (14 Images)

For ease of presentation, we randomly picked 4 images to be used as query images and ranked the other images in the image set, in the order of increasing distance, according to both real and estimated interimage distances.

[Table 1 should be Inserted here]

The result is tabulated in Table 1. The first row for each image in the table gives the ranking based on estimated interimage distances and the second row is based on the real interimage distances. The table shows that for all four query images, the rank ordering based on estimated distances is similar to the one based on real distances. The retrieved images for these queries can be found at
www.cacs.usl.edu/Publications/Raghavan/Images.html. A more detailed analysis is given in [9].

[Table 2 should be Inserted here]

Another approach for evaluating our result is by having an expert identify relevant images. For example, for query image-2, the images that are judged to be relevant are 3, 4, 8, 1, 5, 7, 6, 9, and our retrieval also shows that the closest images are 3, 4, 8, 5, 1, 7, 9, 6. This corresponds to all relevant images being ranked ahead of any non-relevant images. We computed the R_{norm} of the ranking using estimated interimage distance with respect to the real interimage distance and expert provided ranking. The result is very encouraging. The $R_{norm}(\delta^{estreal})$ with respect to the real distance and the $R_{norm}(\delta^{estusr})$ with respect to the expert relevance were computed for these 4 query images and are given in Table 2.

After quantization, the intermediate image representations had 64 low level features. However, after high level features were computed, every image was represented by only six features. This made on-line computation very efficient. A significant factor that contributes to efficiency is that during the computation of the query image feature vectors, we did not have to compute its distance to every image in the database. Instead, we used the training set generated during the initial database population stage.

5.2 EXPT II Results (32 Images)

We experimented with feature space dimensionalities of 1 through 7 and 11 by adjusting the threshold value (tr) in the ComputeFeatureVector procedure. For each of these, the estimated interimage distance between each pair of images were determined.

[Table 3 should be Inserted here]

Treating every image in this set as a query image, we computed rankings of images in the database corresponding to a query image , using the real and the estimated distances. Subsequently we computed the average R_{norm} of the ranking using estimated interimage distance with respect to the real interimage distance for different threshold values. The results are tabulated in Table 3. This table shows the number of features and the corresponding average R_{norm} values for all the queries. It is seen that the R_{norm} value improves when the number of high-level features increases. The increase is substantial at low dimensionalities up to 5. Hence, given an application, we can adapt the computational complexity to the required retrieval accuracy.

6 Conclusion

We have proposed a generic and efficient automatic CBIR architecture. Results for EXPT I show that the dimensionality of the feature space can be substantially reduced (64 to 6), while preserving real interimage distances. R_{norm} of estimated interimage distance with respect to the real interimage distance for all the query images is better than 0.86. Our on-line computation for a given query image is substantially reduced as it needs to compute the real distances only between the query image and the images in the training sample. Also, the retrieval accuracy can be further improved by adjusting the threshold values as we have seen in EXPT II.

References

[1] V. Gudivada and V. Raghavan, "Content-Based Image Retrieval Systems," *IEEE Computer*, vol. 28, no. 9, pp. 18–22, 1995.

[2] S. K. Choubey and V. V. Raghavan, "Generic and fully automatic content-based image retrieval using texture," in *International Conference on Imaging Science, Systems, and Applications (CISST'97)*, (Las Vegas, Nevada), pp. 228–237, June-July 1997.

[3] S. K. Choubey and V. V. Raghavan, "Generic and fully automatic content-based image retrieval architecture," in *10th International Symposium on Methodologies for Intelligent Systems*, (Charlotte, North Carolina), October 1997. (to appear).

[4] S. F. Chang, A. Eleftheriadis, and D. Anastassiou, “Development of Columbia’s Video on Demand Testbed,” *International Journal of Image Communication- Signal Processing*, vol. 8, no. 3, pp. 191–207, 1996.

[5] C. Faloutsos *et al.*, “Efficient and Effective Querying by Image Content,” *Journal of Intelligent Information Systems*, vol. 3, no. 3, pp. 231–262, 1994.

[6] X. Wan and C. C. J. Kuo, “Color Space Quantization for Image Retrieval,” in *SPIE:Storage and Retrieval for Still Image and Video Database’96*, (SanJose California), February 1996.

[7] L. Goldfarb, “A Unified Approach to Pattern Recognition,” *Pattern Recognition*, vol. 17, no. 5, pp. 575–582, 1984.

[8] L. Goldfarb, “A New Approach to Pattern Recognition,” in *Progress in Machine Intelligence and Pattern Recognition, eds: L.N. Kanal and A. Rosenfeld*, vol. 2, North Holland Publishing Company, 1985.

[9] S. K. Choubey, *Generic and Fully Automatic Content-Based Image Retrieval Using Color Shape and Texture*. PhD thesis, University of Southwestern Louisiana Lafayette LA, 1997.

Image ID	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*	*R8*	*R9*	*R10*	*R11*	*R12*	*R13*	*R14*
$\mathrm{Z}(Image-2)$	2	3	4	8	5	1	7	9	6	10	12	14	13	11
$\Pi(Image-2)$	2	3	4	8	5	1	7	9	6	10	14	12	13	11
$Z(Image-4)$	4	7	5	1	8	2	9	12	14	3	10	6	13	11
$\Pi(Image-4)$	4	7	5	1	8	2	9	14	3	12	10	6	13	11
$Z(Image-11)$	11	7	12	8	9	6	14	13	10	5	4	3	1	2
$\Pi(Image-11)$	11	7	12	8	9	14	6	13	10	5	4	1	3	2
$Z(Image-14)$	14	9	10	12	6	7	13	4	11	1	8	5	3	2
$\Pi(Image-14)$	14	9	10	6	12	7	13	4	11	1	8	5	2	3

Table 1: Comparison of Ranking (Z =Estimated Distance, Π =Real Distance)

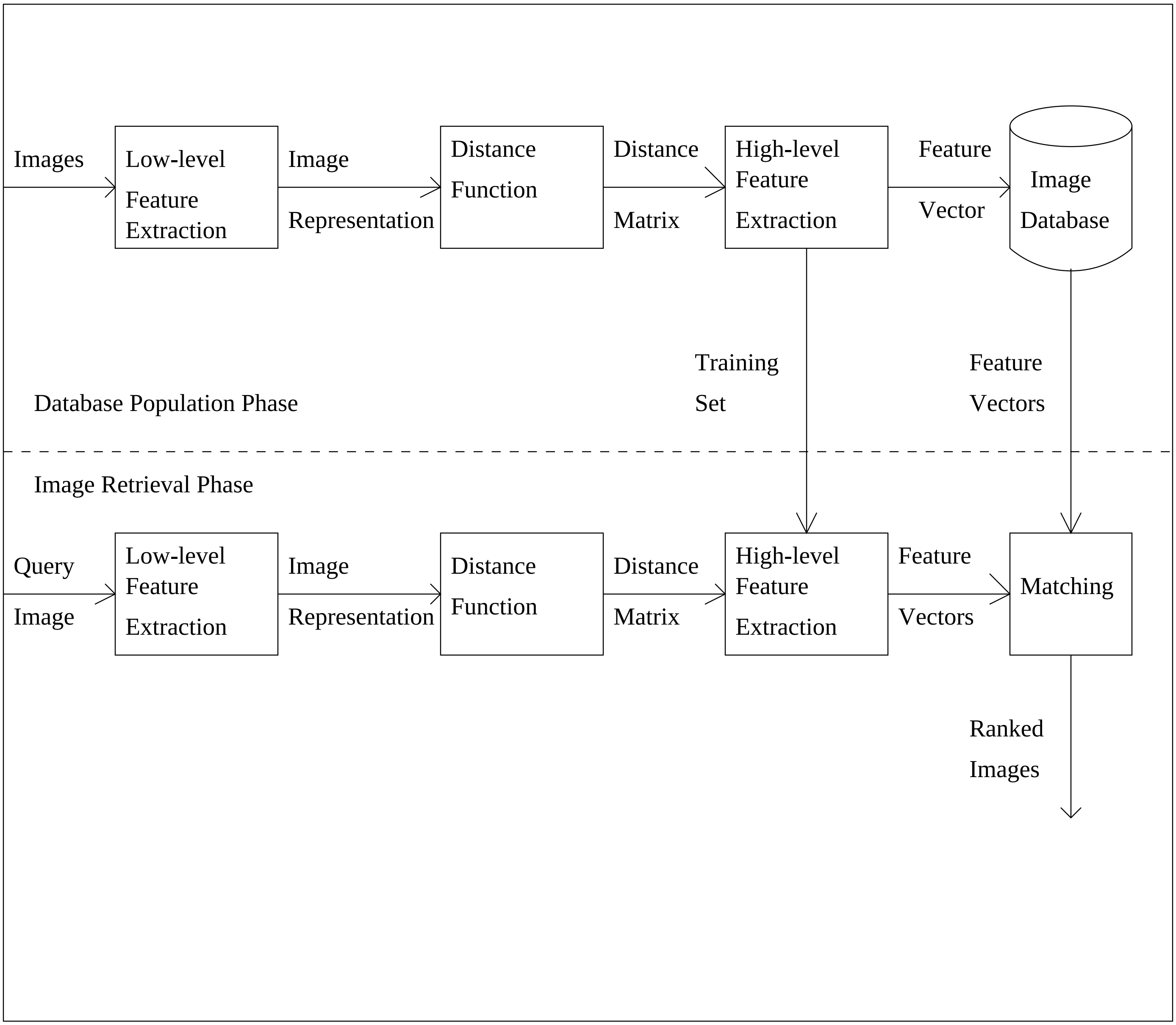

Figure 1: An Architecture of the Proposed System

Query ID	R_{norm}	
	$R_{norm}(\delta^{estreal})$	$R_{norm}(\delta^{estusr})$
Image-2	0.989	0.93
Image-4	0.98876	0.97
Image-11	0.98876	0.97
Image-14	0.98876	0.81

Table 2: R_{norms} for Selected Queries.

Number of High Level Features	R_{norm}
1	0.5874
2	0.6593
3	0.7588
4	0.8036
5	0.8568
6	0.8729
7	0.8888
11	0.93498

Table 3: R_{norm} for EXPT II.